REMARKS

Claims 1-39 are pending. Claims 1-20 have been canceled.

Claims 22-27, 28, 29, 35-39 stand objected to.

Claims 21, 26, 30 and 32 stand rejected under 35 USC §112, second paragraph, as being allegedly indefinite for failing to particularly point out and distinctly claim the subject matter that the Applicant regards as the invention.

Claims 21-26 stand rejected under 35 USC §102(b) as being allegedly anticipated by Shimada (US 5,948,112).

Claims 27-39 stand rejected under 35 USC §103(a) as being allegedly unpatentable over Shimada (US 5,948,112) in view of Carter (US 5,845,331).

Changes in the Claims:

Claims 21-39 have been amended in this application to further particularly point out and distinctly claim subject matter regarded as the invention.

In particular, the claims have been amended to correct translation errors from the priority French application.

Claims Objections

Claim 26 has been objected to because of informalities. Claim 26 has been amended to replace "even though is has usually swapped" to "even though it has usually swapped."

Claims 22-27, 28, 29, 35-39 have been amended to depend as follows:

- Claims 22-27 now depend on claim 21.

- Claims 28 and 29 now depend on claim 27.

- Claims 35-39 now depend on claim 34.

Rejection under 35 USC §112, second paragraph – claims 21 and 26

Claims 21 and 26 stand rejected under 35 USC §112, second paragraph, as being allegedly indefinite for failing to particularly point out and distinctly claim the subject matter that the Applicant regards as the invention. This rejection is respectfully traversed.

MPEP §2171 identifies two separate requirements: (1) the claims must set forth the subject matter that applicants regard as their invention; and (2) the claims must particularly point out and distinctly define the meets and bounds of the subject matter that will be protected by the patent grant. A lack of antecedent basis may be found if a claim is "indefinite" because "it contains words or phrases whose meaning is unclear"; see MPEP §2173.05(e).

The Office Action alleges that claims 21 and 26 are generally narrative and indefinite. Claims 21 and 26 have been amended to further particularly point out and distinctly claim the subject matter that the Applicant regards as the invention.

The Office Action alleges the term "for example" in claims 21 and 26 renders the claim indefinite. Claims 21 and 26 have been amended to delete the term "for example."

The Office Action alleges that the phrase "two virtual sequences located on a single physical sequence are multiplexed in time" in claim 21 lacks antecedent basis. Claim 21 has been amended accordingly to replace "two virtual sequences" with "a first and second virtual channel".

The Office Action alleges that the phrase "variables/data to be voted" in claim 30 lacks antecedent basis. Claim 30 has been amended accordingly to delete the term "variables".

The Office Action alleges the term "the control electronics" in claim 32 renders the claim indefinite. Claim 32 has been amended to delete the term "for example." The claims now meet the statutory requirements.

Rejection under 35 USC §102(b) – claims 21-26

Claims 21-26 stand rejected under 35 USC §102(b) as being allegedly anticipated by Shimada (US 5,948,112). This rejection is respectfully traversed.

A claim must be anticipated for a proper rejection under §102(a), (b), and (e). This requirement is satisfied "only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference"; see MPEP §2131 and *Verdegaal Bros. V. Union Oil*, 814 F.2d 628, 2 USPQ2d 1051 (Fed. Cir. 1984). A rejection under §102(b) may be overcome by showing that the claims are patentably distinguishable from the prior art; see MPEP §706.02(b).

Shimada describes a method for recovering from software fault in a checkpointing and roll back recovery type fault tolerant computing system. Shimada teaches the steps of: detecting faults between particular checkpoints, identifying whether or not the fault is a software fault, recording a faulty portion of an internal state of the system, diagnosing a type of the software fault by using the faulty portion of an internal state of the system, determining a recovery action for the diagnosed type of the software fault, and executing

the recovery action when the internal state of the system rolled back to a checkpoint which was acquired just before occurrence of the detected fault.

In contrast, the presently claimed invention claims:

> Processing procedure for an electronic system subject to transient error constraints, comprising:
>
> multiplexing in time a first and second virtual channels located on a single physical channel for each real time cycle, said real time cycle including an operational cycle of a software task that is executed periodically and continuously,
>
> storing the data resulting from each execution of a virtual channel,
>
> voting these resulting data before using them when the two virtual channels are completed, in order to be able to detect the presence of an error,
>
> canceling the real time cycle in progress in case an error is detected,
>
> reloading a healthy context issued from the previous cycle in case an error is detected,
>
> restarting in case an error is detected, that consists of executing the nominal next cycle starting from the reloaded context.

See Claim 21.

Shimada does not teach or suggest "multiplexing in time a first and second virtual channels located on a single physical channel for each real time cycle". See claim 21. The first and second virtual channels form a "duplex operating on a single physical channel." See specification at page 26, line 4. This feature allows the same software module to be run two successive times on a computer system. See FIG. 11A of the specification. On the other hand, Shimada does not run the same software two successive times but only teaches detecting faults between particular checkpoints and identifying whether or not the fault is a software fault with a fault identification module 103. The "fault identification module 103 is a software module for distinguishing software faults from hardware faults when a fault occurs." See Col. 6, lines 24 of Shimada. Thus, Shimada does not suggest multiplexing two virtual channels on a single physical channel.

Shimana does not teach or suggest "voting these resulting data before using them when the two virtual channels are completed, in order to be able to detect the presence of an error." See claim 21. This allows the computer "to choose or "vote" only the data that are to be output from the computer (the commands), or the data that are used for correction (the context)." See Specification at page 24, line 27- page 25, line 2. On the other hand, Shimada does not describe voting the resulting data when the two virtual channels are completed but teaches "diagnosing a type of the software fault by using the faulty portion of an internal state of the system, determining a recovery action for the diagnosed type of the software fault". The presently claimed application does not require a diagnostic. Shimada requires a diagnostic.

Such a diagnostic can be complicated and difficult because a given type of fault in a microprocessor could generate several other types of faults in a few clock cycles.

A diagnostic as taught by Shimada could yield an amount of computation time which is not suitable for real-time applications such as the one described in the present application.

The diagnostic also requires a lot of work to be developed for every kind of faults, then a lot of work for their validation. As it is well known by those of ordinary skill in the art, fault-tolerant architecture are complex and costly to validate because it is not easy to inject faults and because it is very difficult to be sure that all faults that are injected during the validation phase will be completely representative of faults that they will be encountered into the real environment.

Claim 22

Shimada does not teach or suggest "three error confinement areas (time, software and hardware)". The "confinement areas" are linked to the detection capability of the system and not to its recover capability. The objective of such "confinement areas", as it is well known by those of ordinary skill in the art, is to prohibit a fault which is located into one of the areas, to propagate itself to another area. Thus, the "confinement areas" limit the fault propagation. Thus, Shimada does not teach or suggest "confinement areas".

Claim 23

Shimada does not teach or suggest "an error detection and correction code". As it is well known by those of ordinary skill in the art, "checkpointing" and "roll-back" recovery is a method specifically dedicated to the protection of processing units. On the other hand, "error detection and correction code" is a method specifically dedicated to the protection of memory arrays. Both works completely in different ways. The "error detection and correction code" is a code which is added to the intrinsic memory array. Thus, Shimada does not teach or suggest "an error detection and correction code".

Claim 24

Shimada does not teach or suggest detection /correction in real time cycle. Those of ordinary skills in the art could doubt that the methodology described by Shimada is compliant with real-time applications due to the extensive processing which is required when a fault is detected (identifying whether it is a software or hardware fault, recording

a faulty portion of system internal state, diagnosing a type of the software fault, etc...). Thus, Shimada does not teach or suggest detection /correction in real time cycle.

Claims 25 and 26

Shimada does not teach or describe a "backup or restore context function". On the other hand, the presently claimed application describes the "index change" which allow to activate the swap between "Old " and "New" contexts instantaneously without spending processing time. See page 34, lines 15-17, and page 34, line 25 to page 35, line 3.

The presently claimed invention is, accordingly, distinguishable over the cited reference. In the view of the foregoing, it is respectfully asserted that claims 21-26 are now in condition for allowance.

Rejection of claims 22-26

Claims 22-26 stand rejected under 35 U.S.C. §102(b). These rejections are respectfully traversed for at least the reason that each of the rejected claims ultimately depend on an above-discussed base claim. The arguments set forth above regarding the base claims are equally applicable here. The base claims being allowable, the dependent claims must also be allowable.

Rejection under 35 USC §103(a) – claims 27-39

Claims 27-39 stand rejected under 35 USC §103(a) as being allegedly unpatentable over Shimada in view of Carter. This rejection is respectfully traversed.

Under MPEP §706.02(j), in order to establish a prima facie case of obviousness required for a §103 rejection, three basic criteria must be met: (1) there must be some suggestion or motivation either in the references or knowledge generally available to modify the reference or combine reference teachings (MPEP §2143.01), (2) a reasonable expectation of success (MPEP §2143.02), and (3) the prior art must teach or suggest all the claim limitations (MPEP §2143.03). See In re Royka, 490 F. 2d 981, 180 USPQ 580 (CCPA 1974).

Claims 27-39

Even if Shimada and Carter were to be combined in the manner proposed, the proposed combination would not possess all of the claim limitations of claims 27-39. In particular, Carter teaches a method for checking memory access rights which is only usable with microprocessor designed by the user itself (so, usable only if the user is developing itself an ASIC (Application Specific Integrated Circuit). See Carter at Col. 2, lines 6-7 and lines 41-48 where specific hardware into the microprocessor itself is mentioned. Carter describes the addressing mode. For instance, in FIGS. 1A and 2B each address generated by the microprocessor includes several parameters: pointer tag, permission bits, segment length, address. Those of ordinary skills in the art will realize that such an addressing scheme is not compatible with commercial microprocessor such as INTEL or MOTOROLA families.

In contrast, the presently claimed invention claims checking memory access rights. This method is compatible with the use of commercial microprocessor. The microprocessor programs one or several keys into the hardware "memory access monitoring" device (i.e. the microprocessor writes into registers internally to the device). Then, depending on the value of all the register keys pertaining to this access monitoring device, this device will authorize the microprocessor to access to some segments and prohibit some other ones.

Another advantage of the presently claimed invention is to be able to do in the "access monitoring" device with logical combinations of one or several keys. For instances, the logical combination of the following keys:

- the "task number" key (the memory segment pertained to only one of the applications tasks is accessible at a given time for the microprocessor; See Specification at page 42, lines 8 -13),
- the "virtual sequence number" key (the memory segment pertained to only one of the two virtual sequences ChV#1 and ChV#2 is accessible at a given time for microprocessor; See specification at page 41, line 27 to page 42, line 1).
- the "vote" key (this key indicates that a vote is started and modifies the access rights; See Specification at page 42, lines 4-7).

The keys are used for the "access monitoring" device to authorize the microprocessor to access to memory segments pertained to both virtual channels / sequences (usually prohibited). Obviously, during the vote, memory segments pertained to the two virtual sequences ChV#1 and ChV#2 must be accessed to be able to vote data between ChV#1 and ChV#2 (See specification page 34, lines 6 – 12 and page 37, lines 16

– 19). Thanks to the fact that the key management is done by the "memory access monitoring device", these keys and the logical combinations between them are independent from the chosen microprocessor and thus are compatible with every microprocessors; so, removing the heavy and extremely costly constraint for the user to have to design its own microprocessor as for the Carter.

The presently claimed invention claims the feature that several type of key registers are included in the hardware "memory access monitoring" device, and the feature that logical combinations of these keys allows different configurations of the memory access authorization, nevertheless compatible with the use of commercial microprocessors. None of these limitations are suggested or taught by Carter. On the other hand, Carter requires that each address includes its own authorization bits, which are not compatible with commercial microprocessor.

Applicant therefore submits that the rejection based the Shimada and Carter reference is improper and should be withdrawn. Thus, Applicant submits that claims 27-39 recite novel subject matter which distinguishes over any possible combination of Shimada and Carter.

Claim 31

Even if Shimada and Carter were to be combined in the manner proposed, the proposed combination would not possess all of the claim limitations of claim 31. In particular, Applicant respectfully submit that there is no relationship between the method described by Shimada et al. as "detecting fault between particular checkpoints" and the software vote and its software and hardware protections claimed by the above referenced

application to insure the integrity of the vote which is one of the key stone of this fault-tolerant architecture. These claimed protections are neither taught nor suggested by Shimada et al. because these protections consist of an original set of software and hardware protections as described in the specification:

- Software checks:

- Check of the healthiness of the microprocessor at the beginning of the vote (page 36, lines 18 – 20) by unusual check of the healthiness of the stack pointer and the configuration registers of the electronic control board/card (page 37, lines 6 – 8).

- Software monitoring:

- Inhibition of the cache memories which are very sensitive to singular events like upsets in the space domain (page 37, lines 9 – 10 and page 38, line 25).

- The Vote-Key variable which participates to check the completeness of the vote procedure (cf. from page 37, line 11 to page 38, line 24).

- Computation of a CRC in real-time during the vote itself, this participating to check the completeness of the vote procedure (page 37, lines 20 – 22 and lines 26 – 27, fig. 11B), i.e. it is required to check that a faulty microprocessor does not branch itself in the meddle of the vote code (cf. detailed analysis of the "soft crash" type sequencing errors as described in the fig. 11A and page 36, lines 25 – 28).

- Reinitialisation of the configuration registers of the electronic control board / care to be sure that this board is able to correctly perform its function and then checks them (page 38, lines 4 – 5 and lines 7 – 9), i.e. it is required to check that a faulty microprocessor does not branch itself in the meddle of the vote code (cf. detailed

analysis of the "soft crash" type sequencing errors as described in the fig. 11A and page 36, lines 24 – 28).

- Vote of the transferred tables "Old" and "New" (page 38, lines 18 – 22).

- Hardware monitoring:

- Watchdog (page 36, lines 29 – 30).

- Activate the key of the "memory access monitoring" device indicating voting is being done (page 37, lines 16 – 19), only for the minimum duration (page 37, lines 29 – 30); this key exceptionally authorizing the microprocessor to simultaneously access to both the memory segments associated to the two virtual channels / sequence ChV#1 and ChV#2 (page 42, lines 4 – 7).

If the microprocessor is able to successfully pass all these software and hardware protections, one can conclude that there is a strong probability that the microprocessor is healthy during the vote. As is known by those ordinary skilled in the art, it is very difficult to be sure of the healthiness of a microprocessor. To reach this strong probability, all the cross-checks reminded hereabove are proposed thanks to the detailed analysis of the "soft crash" type sequencing errors concerning the vote as described in the fig. 11A (cf. page 36, lines 24 – 28).

Applicant therefore submits that the rejection based the Shimada and Carter reference is improper and should be withdrawn. Thus, Applicant submits that claim 31 recites novel subject matter which distinguishes over any possible combination of Shimada and Carter.

Claim 33

The arguments set forth above regarding claim 21 are equally applicable here. Several features of the above referenced application are the real-time functioning and the simplicity / cheapness of the concept. In contrast, Shimada proposes a complex diagnostic method which generate difficulties with the real-time feature of the application and with the complexity and cost of the validation phase and injection tests during satellite integration.

Applicant therefore submits that the rejection based the Shimada and Carter reference is improper and should be withdrawn. Thus, Applicant submits that claim 33 recites novel subject matter which distinguishes over any possible combination of Shimada and Carter.

Claim 34

The arguments set forth above regarding claim 29 are equally applicable here. As indicated earlier, the claimed limitation of "hardware device to check access rights and key pointers" is compatible with the use of every commercial microprocessor which is an original feature thereby removing the heavy and extremely costly constraint for the user to have to design its own microprocessor as for the Carter.

As indicated in the arguments regarding claim 31, it is very difficult to be sure in real-time the healthiness of a microprocessor, specifically during "soft crashes" (See Specification from page 23, line 11 to page 24, line 19). So, the following feature: "access for some (memory) segments will only be authorized if there is a strong probability that the microprocessor will be in a good operation state", describes mixing

several protections and not to use a single one. The method proposed by Carter is based on a single protection which is the "permission bits" included into each address (which requires to design its own microprocessor, such a design being not reachable for a great majority of companies due to its extremely high cost). On the other hand, the method proposed by the presently claimed invention is compatible to the use of every commercial microprocessor. For instance, the microprocessor will have access simultaneously to both memory segments ChV#1 and ChV#2 (See specification at page 37, lines 16 – 19) could be seen as a dangerous possibility due to potential fault propagation between the two virtual channels / sequences (such faults could be undetectable because present into both virtual channels); but the microprocessor will have exceptionally access to TAB-Ctxt-New#1 and simultaneously to TAB-Ctxt-New#2 only during the vote procedure with the insurance that it is in good operating state thanks to all the software and hardware protections included in the vote procedure.

It should be reminded that the memory segments used for the storage of the context data are the most important memory segments and so, must be very well protected due to the criticality of their content. Indeed, the context data must be healthy in order to be able to recover from a fault (See specification at page 12, lines 28 – 29). The above referenced application "enable safe storage of critical data" specifically for these segments. The different uses which are made of these segments are the following ones:

- During the "processing phase", the context resulting from the processing are written into TAB-Ctxt-New (cf. page 33, lines 23 – 24 and page 32, lines 3 – 6).

- Then, during the "vote phase", the integrity of the two context segments TAB-Ctxt-New#1 and TAB-Ctxt-New#2 is checked, thanks to the vote.
- Then, if the vote is successful (and there is a lot of hardware and software protections during the vote to insure that the microprocessor is in a good operating state as already seen in point 24 hereabove), an index is changed in order to swap Tab-Ctxt-New into TAB-Ctxt-Old (it is the Tab-Ctxt-Old tables which store the data context for the recovery).

This is possible thanks to the limitations claimed in the presently claimed invention such as "confinement areas" between tasks and between real-time cycles (cf. page 30, line 27 to page 31, line 3) allowing to vote a context for each task during each real-time iteration; like "process tables" (cf. page 31, line 5 – 24) allowing to swap TAB-Ctxt-New and TAB-Ctxt-Old just changing an index; like "secure software vote" and all its hardware and software protections as already seen in item 24; and like "memory access monitoring" device (cf. page 40, line 27) playing a central role with regard to memory protection.

Finally, if the integrity / healthiness of TAB-Ctxt-New is successfully checked, then the transfer of TAB-Ctxt-New towards TAB-Ctxt-Old is almost instantaneously. So because, data has been checked before the swap, and the swap is instantaneously, the probability to have erroneous data into TAB-Ctxt-Old is extremely low.

Accordingly, because the microprocessor never is authorized to write into the memory segment used for TAB-Ctxt-Old storage (it writes only into the segment for TAB-Ctxt-New, cf. page 32, lines 3 – 6), a faulty microprocessor is never capable to corrupt the back-up context TAB-Ctxt-Old.

So, the presently claimed invention enables safe storage of critical data in an original way whereas Carter has only a simple "right access" allowed by "permission bits" into each address and, more, requiring an ASIC specific microprocessor.

Applicant therefore submits that the rejection based the Shimada and Carter reference is improper and should be withdrawn. Thus, Applicant submits that claim 34 recites novel subject matter which distinguishes over any possible combination of Shimada and Carter.

Claims 35-39

Claims 35-39 stand rejected under 35 U.S.C. §103. These rejections are respectfully traversed for at least the reason that each of the rejected claims ultimately depend on an above-discussed base claim. The arguments set forth above regarding the base claims are equally applicable here. The base claims being allowable, the dependent claims must also be allowable.

Furthermore, the presently claimed invention claims the possibility to have logical combination functions for the keys whereas Carter teaches only a simple "right access" allowed by "permission bits" into each address and, more, requiring an ASIC specific microprocessor.

Storage by pair is another original feature taught by the above referenced application: a first "pair level" is constituted by the tables associated to the first virtual channel / sequence ChV#1 and the tables associated to the second one ChV#2 (cf. page 31, lines 8 – 10); and included into this first pair, a second "pair level" is constituted by "Old" and "New" parts (cf. page 18, lines 28 – 30), so having for instance ChV#1-New

and ChV#2-New tables (e.g. TAB-Ctxt-New#1 and TAB-Ctxt-New#2, page 33, line 24 and page 34, line 4). And an important feature resides in the fact that each pair of tables have its own authorization mode. Indeed, for instance, if the microprocessor is working on TAB-Ctxt-New#1, then read and write access to all ChV#2 memory segments is prohibited; at that moment, if an error corrupt the index (ChV#1 is changed in ChV#2 due to this error), so the microprocessor will try to access to the ChV#2 memory and this unauthorized access will be detected as an error, generating a recovery mode. In other words, data are stored by pair, but each element of the pair have its own authorized mode. The simple addressing mode with an offset as the one disclosed by Carter et al. e.g. at figure 2B allows working by pair but, unfortunately, the offset mechanism (allowing to work by pair) is not correlated with the permission bits as for the addressing mechanisms taught by the above referenced application. It should be noted that the method disclosed by Carter et al. is particularly applicable in a multiprocessor environment which is not the case in the space domain.

Conclusion

For all of the above reasons, applicants submit that the amended claims are now in proper form, and that the amended claims all define patentable subject matter over the prior art. Therefore, Applicants submit that this application is now in condition for allowance.

Request for allowance

It is believed that this Amendment places the above-identified patent application into condition for allowance. Early favorable consideration of this Amendment is earnestly solicited. If, in the opinion of the Examiner, an interview would expedite the prosecution of this application, the Examiner is invited to call the undersigned attorney at the number indicated below.

Respectfully submitted,
THELEN REID & PRIEST LLP

Dated: June 4, 2004

Thierry K. Le
Reg. No. 49,097

Thelen Reid & Priest LLP
P.O. Box 640640
San Jose, CA 95164-0640
(408) 282-1810 Direct
(408) 292-5800 Main
(408) 278-8210 Fax